JTC Resource Bulletin

Leveraging Open Source Resources

Version 1.0
Adopted 19 April 2019

Abstract

The world runs on software. Both public and private sector organizations are increasingly utilizing non-proprietary open source code. Courts have yet to significantly leverage open source options and fully participate in shared repositories. Leveraging open source solutions can help courts expedite the process of adopting innovations that would otherwise be too costly to pursue. Utilizing open source code and repositories will create opportunities for collaboration, bringing the best ideas in the broader court community to play in many individual court organizations.

Document history and version control

Version	Date Approved	Approved by	Brief Description
1.0	4/19/2019	JTC	Release document

Acknowledgments

This document is a product of the Joint Technology Committee (JTC) established by the Conference of State Court Administrators (COSCA), the National Association for Court Management (NACM) and the National Center for State Courts (NCSC).

JTC Mission:
To improve the administration of justice through technology

Joint Technology Committee

Compiled and written by Lise Embley, Technical Writer, National Center for State Courts

Additional Contributors

Vesna Milojevic, NCSC Fellow

Table of Contents

Introduction to open source software and repositories

The world runs on software. As courts increasingly rely on software (applications) to manage the day-to-day business of case management, calendaring, jury management, payment processing, document management, etc., courts must have robust applications suited to their unique requirements. To meet their software needs, courts generally utilize proprietary software in the form of customizable off-the-shelf (COTS) products or fully custom solutions that are developed in-house or through external contractors and vendors.

However, proprietary software is not the only option. Both public and private sector organizations are increasingly utilizing non-proprietary *open source* code. Open source software is "shareware" code developed and/or stored in a cloud-based *repository* that facilitates sharing as well as ongoing development.

Some courts have utilized open source code in limited ways, incorporating minor components into in-house development. Others have shared software informally. Some are utilizing repositories (private or public) for software development. However, courts as a whole have yet to significantly leverage open source software and fully participate in shared repositories.

About software

Software consists of instructions in the form of code or scripts that tell computers how to do specific tasks. In the 1960s and 1970s, academics and corporate laboratories customarily shared software freely. The synergy associated with sharing, modifying, and re-sharing code facilitated much of the innovation and rapid development that ushered in the computer era. In the 1980s, things shifted. Source code became a guarded corporate asset protected by license agreements establishing terms of use.[1] Proprietary software was born.

Proprietary software may be developed or purchased "stand alone" or come bundled as part of a device purchase. Users must agree to terms of use with strict limitations. Modifications or customization is specifically prohibited. Robust competition has driven software companies like Microsoft, Oracle, and Adobe to create unique products that power much of the business world and generate billions in revenues. Proprietary software is not going away.

Open source products like WordPress, Magento, LibreOffice (a name which echoes Latin words meaning liberty or freedom), and Ubuntu (an African philosophy of

[1] von Hippel, Eric, and Georg von Krogh. "Open Source Software and the 'Private-Collective' Innovation Model: Issues for Organization Science." *SSRN*, Social Science Research Network, 30 Apr. 2002. MIT Sloan Research Paper No. 4739-09

community, sharing and generosity) are also used by millions of people every day. Wikipedia is not only community-driven content, it is community-developed software, as well. Open source software including the Linux operating system and Apache web server are key technologies that facilitated the development of the web.[2] Innovation today continues to hinge heavily on free and open source software.[3]

> Linux showed the world at large that a new way of working where co-creation, volunteerism, meritocracy and other such non-traditional approaches can produce something powerful. [4]

Participatory culture, or a culture of working collaboratively and sharing, is central to both open source software and repositories. Sharing everything from bikes and cellular networks to sofas and software is trendy and can be transformational. Organizations such as the Free Software Foundation, The Open Org, redhat, and others promote the development community and provide tools and governance that facilitate sharing. Today, both proprietary and open source software are essential components of a thriving software development ecosystem.

About repositories

Broadly speaking, a repository is simply a website where software code is stored. It is the mechanism that facilitates access, sharing, and collaboration. Individuals, businesses, and government entities commonly utilize software repositories to manage software development efforts.

Repositories can be public or private; some offer both options. Anyone can view, utilize, and collaborate on open source code housed in a public repository. A private repository can only be viewed and accessed by the owner of the repository and any parties to whom the owner grants permission.[5]

Software developed in a private repository may be released to a public repository once it is complete. Hosting a project on a public repository facilitates sharing on a broader scale. Since the intent of open source is sharing, open source software is usually housed in a public repository. Beyond sharing, a repository facilitates collaboration through version control mechanisms and other collaboration tools.

[2] Fulton, Scott. “How Linux Conquered the Data Center.” *Data Center Knowledge*, Informa USA, Inc., 6 Mar. 2017.

[3] See Ebert, Christof. "Open Source Software in Industry," in *IEEE Software*, vol. 25, no. 3, May/June 2008. pp. 52-53.

[4] Kepes, Ben. “Open Source Is Good And All, But Proprietary Is Still Winning.” *Forbes*, Forbes Magazine, 8 Oct. 2013

[5] For more information see https://help.github.com/articles/about-repositories/

Repository platforms abound. GitHub, SourceForge, Bitbucket, GitLab, Launchpad and others offer a variety of unique features. GitHub, which houses approximately 85 million repositories, is the largest and best-known platform for open source projects and offers both public and private repositories.[6] It is free for public and open source projects; paid plans are available for private repositories. Support and security features differ depending on the user's choice of repository. Many federal, state, and local government agencies including the FDA, GSA, NIH, NASA, Centers for Disease Control and the New Orleans Office of Performance and Accountability use GitHub.[7]

Weighing the implications

There are benefits as well as challenges to utilizing open source software. Courts should carefully consider both.

One of the obvious advantages of open source may be cost since the basic code is free. Customization and implementation costs would be similar to those aspects of other in-house COTS or custom efforts, but overall costs may be lower. Additional savings may come from reduced administrative costs and increased integration options that proprietary software vendors may limit.

Inviting other courts and the general public to collaborate through a public repository has the potential to broaden development resources and improve software quality. Collaboration could help courts better identify software security issues, resolve software problems faster, and create a software base standard while affording individual courts the flexibility to make modifications to address their own unique needs. In theory, the more people that work on and test a product, the better the software should be.

An introduction to the 2016 Federal Source Code Policy[8] calls attention to one of the specific financial advantages of sharing source code: avoiding duplicative development efforts that waste taxpayer money. There are thousands[9] of state, county, local, and specialty courts throughout the United States, all with fairly similar business and data requirements. Duplicative development efforts abound. Cross-jurisdiction collaboration could lower overall development and maintenance costs.

Open source software does not come with a "guaranteed service level" or technical support. Individual courts utilizing the same open source software must each allocate adequate support resources. Purchasing processes present a significant barrier. There is likely no simple mechanism for considering an open source option as a competitor to

[6] For more information, see https://github.com/open-source

[7] See https://github.com/collections/government for the growing list of government-specific apps available.

[8] United States, Congress, Office of Management and Budget, et al. "Federal Source Code Policy." 8 Aug. 2016. sourcecode.cio.gov/#fn3

[9] *BNA's Directory of State and Federal Courts, Judges, and Clerks 2019: A State-by-State and ... Federal Listing* lists more than 2,300 state and federal courts.

an RFP response. Courts may not be aware of open source options because open source software is not marketed. Managing an open source project would require a high level of communication across organizational boundaries within as well as outside the court, and court leaders and managers with the unique skills needed to nurture collaboration.

In spite of the challenges, the potential benefits of open source should warrant investigating options. A summary of some of the advantages and disadvantages follows.

Advantages	Disadvantages
Initial cost Free. Courts would incur costs to customize and implement software, but those costs are often lower than with proprietary software. Additional savings may come from reduced administrative costs relating to managing user licenses. **Hardware** Often platform and device agnostic.[10] Delivering a consistent experience across devices could reduce costs associated with narrow hardware and/or operating system requirements. **Component flexibility** Potential for integration with other court software. Vendors may limit integration options to force "turn key" solutions, essentially holding clients "hostage" to their development schedule and pricing. **Customization** Unique requirements could potentially be addressed by internal IT resources. **Reliability** A robust development community will work to continuously improve the software.	**Future cost** The community that developed the software is not obligated to update or support it. Problems may emerge. Adaptations may be required to address changes in court processes. These and other scenarios could potentially require additional development resources. **Technical support** No 800-number or 24/7 support. Courts must ensure adequate in-house support resources. **User experience** May be less user-friendly since developers may favor functionality over user experience. **Security** With source code exposed to public view, malicious users may discover and exploit vulnerabilities. **Purchase process** There is no unified entity to compete for work. **Court culture** Requires culture of openness and sharing that may be challenging to achieve.

[10] *Platform agnostic* refers to software that runs well on multiple types of devices and operating systems (MS Windows, Mac OS, Unix, Linux, iOS, Android, etc.).

When the market lacks quality commercial solutions, open source may be a particularly good option even if costs would equal or exceed those of a commercial or in-house solution. The long-term benefits of collaboration with other jurisdictions may justify equal or even somewhat higher cost.

Licensing considerations

Open source software is created deliberately through a license. Typically, open source licenses permit users to modify the source code, make derivative works, and redistribute the software with certain caveats. Those who share software can generally assign the type of license that best reflects how protective or permissive they wish to be with the code they are sharing. In some instances, utilizing a public repository may automatically assign a sharing license.

In general terms, there are three major types of open source software licenses with nicknames that cleverly hint at differences as compared to traditional copyright:

Permissive “copycenter”	Similar to public domain. Gives permission to use or modify software with minimal requirements on redistribution. Code that begins as open source with a permissive license can become part of an organization’s proprietary code as long as the original developer is acknowledged (“attribution”).
Strongly Protective “copyleft” or reciprocal	Allows anyone to modify and freely distribute other versions with the stipulation that the same rights carry forward into any derivative works. Protects the rights of the original developer(s) and prevents future iterations from ever becoming propriety. Does not allow the addition of restrictions that could be required if copyleft code is combined with other licensed code.
Weakly Protective	A balance of attributes between permissive and strongly protective licenses. Allows software to be combined into a larger proprietary program without becoming proprietary. Users must explain changes made to the software and include the text of the original license in software that has been modified.

Nearly all open source licenses include language explaining that the software comes without a warranty and that the developer cannot be held liable for damages.

Unless protected by copyleft license, open source software could potentially be incorporated into a private-sector proprietary solution. This issue should be addressed through licensing to prohibit proprietary software vendors from re-packaging open source software and selling it to courts that are unaware of the free open source options. Licensing would also make it possible for private-sector enhancements to be shared back with the court.

Courts will need to assess how “open” they want to make their solution. The Open Source Initiative[11] has in-depth information about License options and Standards. Prior to using or creating open source software, courts should consider any unique factors that would influence licensing selection. Court managers should ensure software licensing language is reviewed by a lawyer specializing in Intellectual Property.

Managing an open source project

Like any software project, an open source initiative requires a plan that includes design, development, documentation, and implementation. Some aspects of an open source project are somewhat different than other internal development efforts. The following are likely differences, and there may be others.

Development resources

Undertaking an open source project will require at least some level of in-house open source expertise. Evaluate your IT talent pool to identify any developers that have experience with open source programming. Since open source is not commonly utilized by courts, it is unlikely that staff who do have open source experience will have obtained that experience in a court-related organization. Any open source experience can be leveraged to help with a court-specific open source project.

If your court does not currently have in-house open source expertise, work with Human Resources to add that as a skill requirement for the next open IT position. A contractor may be able to fill that role until the court can bring that expertise in-house.

With the help of your open source programming resource, review the court’s business need and discuss what the program needs to do. A developer will be the best person to help determine whether an open source or third-party licensed solution would be better suited to fulfill the software need. The developer may also provide valuable feedback on which open source repository should be used and why.

Most court administrators will not be knowledgeable about potential repositories and will need to be able to fully rely on a developer’s expert opinion. Once they receive that opinion then the administrator should seek legal counsel to review and approve open source licensing for the chosen repository.

[11] The Open Source Initiative (OSI) is a 501(c)3 tax-exempt status public benefit corporation. Founded in 1998, the OSI are “stewards of the Open Source Definition (OSD) and the community-recognized body for reviewing and approving licenses as OSD-conformant.” See https://opensource/about.

Documentation

Open source project documentation generally includes a **Readme** that explains the project to new contributors and community members. **Contributing documents** explain what contributions are needed and how the process works. Some projects also have a **Code of Conduct** document that establishes rules for behavior. Courts participating in open source development should expect to both utilize and contribute to project documentation.

Contributor roles

Depending on the size of the project, an open source development effort includes a variety of individuals with different roles. Each project can include one or more authors, owners, maintainers, contributors, community members, and even subcommittees focusing on different tasks.[12] These individuals do not need to know each other, work for the same organization, or work from the same location.

Author	Creates the project.
Owner	Holds exclusive administrative rights to distribute a project.
Maintainer	Fixes bugs, answers questions, creates tutorials and documentation. Maintainers are key to the long-term success of a project.
Contributor	Anyone who has worked on the project.
Community	Individuals/organizations that use the project.

Today, millions of developers contribute to open source development efforts.

Governance

Governance is especially important with open source projects. Contributors could include people from organizations with very similar or vastly different priorities. Some governance considerations are uniquely important in an open source project, e.g., how the court will make decisions about the project and how the software will be maintained going forward. Like open source licensing terms, open source governance approaches have clever descriptors:

BDFL	"Benevolent Dictator for Life" – a single individual with the authority to make major project decisions.

[12] See https://opensource.guide/how-to-contribute/#what-it-means-to-contribute

Meritocracy	Outstanding project contributors are given a formal decision-making role. One of the key ideals of open source is that "good ideas can come from anyone or anywhere, and the best ideas win."[13]
Liberal Contribution	Includes as many community perspectives as possible in decisions about the project.

Court culture

Open source will require a different kind of leadership. The processes, skills, and mindset used to oversee traditional in-house development or customization may stymie essential open source activities like building a developer community and working collaboratively both within and outside the court. Taking on an open source initiative could help nudge a court toward better information sharing within the court organization. How courts might share governance authority is as-yet uncharted territory in open source development.

The open source landscape for courts

Open source code can help courts address day-to-day business requirements. Leveraging open source solutions can help courts expedite the process of adopting innovations that would otherwise be too costly to pursue. Sharing facilitates innovation that could benefit both the courts that adopt and modify shared code as well as the court that shared it initially. Collaboration can bring the best ideas from the broader court community into many individual court organizations.

With the shared "pain points" of public sector budgets, courts of all sizes recognize the need to more fully leverage existing work. Courts that have developed their own software solutions are often willing to share their work with other courts. Informally, courts may simply exchange code through the use of transfer devices including USB drives and .zip files, or by storing and sharing their source code via private repository. Almost any custom software in existence or under development today could be licensed and shared as open source software.

The US government has officially embraced open source. In late 2016, the White House announced code.gov, a platform hosting site offering both "government-wide reuse" code and open source code from across the spectrum of federal agencies. Thousands of repositories within code.gov share code and invite participation in development tasks. The Agency Compliance page gives a scorecard of each agency's level of compliance with the Federal Source Code Policy. As the report demonstrates, federal agencies are

[13] Alexander, DeLisa. "Meritocracy 2.0: A Framework for Decision-Making." Opensource.com, RedHat, Inc., 1 June 2016, opensource.com/open-organization/16/6/presenting-framework-meritocracy.

somewhat uneven in their adoption of open source. But progress is measurable and public.

Where vendors have been slow to offer innovative products to fully address evolving business requirements in the justice space, individual development efforts are underway. In Utah, for example, the AOC has successfully implemented a custom ODR solution in three counties and will be expanding to other courts in coming months. Efforts are underway in Alaska and Hawaii to build the foundation for a legal services portal. Minnesota has offered to share its MyMNConservator software with other states. Other states have expressed a willingness to share, but a mechanism similar to code.gov for sharing by state and local agencies is not yet in place.

As more courts both draw on and contribute to open source software, the overall quality of court software will increase while software costs for many courts may decrease. In many instances, the advantages of utilizing open source software will significantly outweigh any potential challenges and can help foster innovation.

For more information about current open source initiatives involving courts,
contact NCSC at technology@ncsc.org.

Appendix A: Taking Action

Ready to look at sharing or leveraging shared code in your court? Use the following possible actions as a checklist to guide discussion.

Suggested Court Actions	Action Level
☐ Inventory the full breadth of the court's current web, desktop, and mobile applications and integrations with partner agencies.	Basic
☐ Identify custom code that may be worth sharing.	Basic
☐ Consult with in-house development resources to identify open source expertise.	Basic
☐ Work with Human Resources to add open source experience as a required qualification when hiring developers going forward.	Basic
☐ With the help of an in-house open source programmer, identify projects where open source could be considered in addition to RFP responses.	Basic
☐ Identify business needs that are currently unmet that might be addressed through open source options and innovations.	Intermediate
☐ With the help of your in-house resource, evaluate features and benefits of potential repositories (private and public).	Intermediate
☐ Consider any court-specific factors that would influence licensing selection.	Intermediate
☐ Obtain legal assistance to determine the OSI-approved license the court would utilize when sharing software.	Advanced
☐ Establish guidelines for how a shared project would be managed and governed.	Advanced
☐ Consider what would be required to host an open source solution. Explore third party cloud and on-premise hosting options. If hosting on-premise, evaluate operating systems (Linux, Windows, MacOS).	Advanced
☐ Consider interface requirements to integrate open source programs with legacy proprietary systems.	Advanced